

AUS920030604US1

RECEIVED
CENTRAL FAX CENTER

JUN 29 2006

AMENDMENT

Please amend the above-identified application as follows:

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the above-identified application:

Listing of Claims:

1. (Currently Amended) A method of content addressable data storage and compression for semi-persistent computer memory comprising:

~~providing a chunk of data comprising a quantity of input data;~~

~~retrieving a memory block from semi-persistent computer memory;~~

~~searching for a segment of the chunk that matches the memory block; and~~

searching at a repeating memory interval through a search section of a chunk for a segment of the chunk that matches a memory block from computer memory, including: calculating a weak checksum for the memory block; calculating rolling weak checksums for segments of the search section of the chunk; comparing the rolling weak checksums for the segments with the checksum for the memory block; and if a segment is found with a rolling weak checksum equal to the weak checksum of the memory block; calculating a strong checksum for the memory block; calculating a strong checksum for the segment with the matching rolling weak checksum; comparing the strong checksum of the memory block and the strong checksum for the segment with the equal rolling weak checksum;

AUS920030604US1

determining that the search has found a segment having contents that match the contents of the memory block if the strong checksum of the memory block and the strong checksum for the segment with the matching rolling weak checksum are equal;

if a matching segment is found:

discarding the matching segment;

providing a retrieval key for the memory block as a retrieval key for the matching segment;

identifying an unmatched portion of the chunk that does not match the memory block;

identifying a free memory block of a file system;

storing the unmatched portion semi-persistently in the free memory block; and

providing a retrieval key for the unmatched portion.

2. (Original) The method of claim 1 wherein a free memory block of a file system has a block size at least as large as a maximum memory block size.
3. (Original) The method of claim 1 wherein storing the unmatched portion semi-persistently in the free memory block comprises storing the unmatched portion without recording the use of the free memory block in the file system.
4. (Original) The method of claim 1 wherein:

AUS920030604US1

identifying a free memory block of a file system comprises reading a block identification from a free block list of a file system, and

storing the unmatched portion semi-persistently in the free memory block comprises leaving the block identification unchanged in the free block list of the file system.

5. (Cancelled)
6. (Cancelled)
7. (Original) The method of claim 1 wherein storing the unmatched portion of the chunk comprises storing the unmatched portion of the chunk as a new memory block having a memory block size equal to the size of the unmatched portion of the chunk.
8. (Original) The method of claim 1 wherein searching for a segment of the chunk that matches the memory block fails to find a matching segment, the method further comprising repeatedly carrying out the following steps for all memory blocks in computer memory until a matching segment is found:

retrieving a next memory block from computer memory; and

searching for a segment of the chunk that matches the next memory block.
9. (Original) The method of claim 8 wherein no matching segment is found in any memory block in computer memory, the method further comprising:

identifying a free memory block of a file system;

AUS920030604US1

storing a search section of the chunk semi-persistently in the free memory block;
and

providing a retrieval key for the search section of the chunk.

10. (Original) The method of claim 8 wherein storing a search section of the chunk comprises storing the search section of the chunk as a new memory block having a memory block size equal to the size of the search section of the chunk.

11. (Original) The method of claim 8 wherein providing a retrieval key for a search section of a chunk comprises:

calculating a weak checksum for the search section of the chunk; and

calculating a strong checksum for the search section of the chunk.

12. (Original) The method of claim 1 further comprising:

receiving a retrieval key;

identifying a memory block in dependence upon the retrieval key;

retrieving the identified memory block; and

verifying the contents of the memory block.

13. (Original) The method of claim 12 wherein the retrieval key for the memory block comprises a unique key calculated with an algorithm that generates a unique key from the contents of a memory block, and verifying the contents of the memory block further comprises:

AUS920030604US1

calculating a new key for the memory block with the same algorithm; and

comparing the retrieval key and the new key.

14. (Currently Amended) A system for content addressable data storage and compression for semi-persistent computer memory comprising:

~~means for providing a chunk of data comprising a quantity of input data;~~

~~means for retrieving a memory block from semi-persistent computer memory;~~

~~means for searching for a segment of the chunk that matches the memory block;~~

means for searching at a repeating memory interval through a search section of a chunk for a segment of the chunk that matches a memory block from computer memory, including means for: calculating a weak checksum for the memory block; calculating rolling weak checksums for segments of the search section of the chunk; comparing the rolling weak checksums for the segments with the checksum for the memory block; and if a segment is found with a rolling weak checksum equal to the weak checksum of the memory block: calculating a strong checksum for the memory block; calculating a strong checksum for the segment with the matching rolling weak checksum; comparing the strong checksum of the memory block and the strong checksum for the segment with the equal rolling weak checksum;

means for determining that the search has found a segment having contents that match the contents of the memory block if the strong checksum of the memory block and the strong checksum for the segment with the matching rolling weak checksum are equal;

means for discarding a matching segment;

AUS920030604US1

means for providing a retrieval key for the memory block as a retrieval key for the matching segment;

means for identifying an unmatched portion of the chunk that does not match the memory block;

means for identifying a free memory block of a file system;

means for storing the unmatched portion semi-persistently in the free memory block; and

means for providing a retrieval key for the unmatched portion.

15. (Original) The system of claim 14 wherein a free memory block of a file system has a block size at least as large as a maximum memory block size.

16. (Original) The system of claim 14 wherein means for storing the unmatched portion semi-persistently in the free memory block comprises means for storing the unmatched portion without recording the use of the free memory block in the file system.

17. (Original) The system of claim 14 wherein:

means for identifying a free memory block of a file system comprises means for reading a block identification from a free block list of a file system, and

means for storing the unmatched portion semi-persistently in the free memory block comprises means for leaving the block identification unchanged in the free block list of the file system.

AUS920030604US1

18. (Cancelled)

19. (Cancelled)

20. (Original) The system of claim 14 wherein means for storing the unmatched portion of the chunk comprises means for storing the unmatched portion of the chunk as a new memory block having a memory block size equal to the size of the unmatched portion of the chunk.

21. (Original) The system of claim 14 wherein means for searching for a segment of the chunk that matches the memory block fails to find a matching segment, the system further comprising means for repeatedly carrying out the following steps for all memory blocks in computer memory until a matching segment is found:

retrieving a next memory block from computer memory; and

searching for a segment of the chunk that matches the next memory block.

22. (Original) The system of claim 21 further comprising:

means for identifying a free memory block of a file system;

means for storing a search section of the chunk semi-persistently in the free memory block; and

means for providing a retrieval key for the search section of the chunk.

23. (Original) The system of claim 21 wherein means for storing a search section of the chunk comprises means for storing the search section of the chunk as a new memory block having a memory block size equal to the size of the search section of the chunk.

AUS920030604US1

24. (Original) The system of claim 21 wherein means for providing a retrieval key for a search section of a chunk comprises:

means for calculating a weak checksum for the search section of the chunk; and

means for calculating a strong checksum for the search section of the chunk.

25. (Original) The system of claim 14 further comprising:

means for receiving a retrieval key;

means for identifying a memory block in dependence upon the retrieval key;

means for retrieving the identified memory block; and

means for verifying the contents of the memory block.

26. (Original) The system of claim 25 wherein the retrieval key for the memory block comprises a unique key calculated with an algorithm that generates a unique key from the contents of a memory block, and means for verifying the contents of the memory block further comprises:

means for calculating a new key for the memory block with the same algorithm;
and

means for comparing the retrieval key and the new key.

27. (Currently Amended) A computer program product for content addressable data storage and compression for semi-persistent computer memory comprising:

AUS920030604US1

a recording medium;

~~means, recorded on the recording medium, for providing a chunk of data comprising a quantity of input data;~~

~~means, recorded on the recording medium, for retrieving a memory block from semi-persistent computer memory;~~

~~means, recorded on the recording medium, for searching for a segment of the chunk that matches the memory block;~~

means, recorded on the recording medium, for searching at a repeating memory interval through a search section of a chunk for a segment of the chunk that matches a memory block from computer memory, including means, recorded on the recording medium, for: calculating a weak checksum for the memory block; calculating rolling weak checksums for segments of the search section of the chunk; comparing the rolling weak checksums for the segments with the checksum for the memory block; and if a segment is found with a rolling weak checksum equal to the weak checksum of the memory block: calculating a strong checksum for the memory block; calculating a strong checksum for the segment with the matching rolling weak checksum; comparing the strong checksum of the memory block and the strong checksum for the segment with the equal rolling weak checksum;

means, recorded on the recording medium, for determining that the search has found a segment having contents that match the contents of the memory block if the strong checksum of the memory block and the strong checksum for the segment with the matching rolling weak checksum are equal;

means, recorded on the recording medium, for discarding a matching segment;

AUS920030604US1

means, recorded on the recording medium, for providing a retrieval key for the memory block as a retrieval key for the matching segment;

means, recorded on the recording medium, for identifying an unmatched portion of the chunk that does not match the memory block;

means, recorded on the recording medium, for identifying a free memory block of a file computer program product;

means, recorded on the recording medium, for storing the unmatched portion semi-persistently in the free memory block; and

means, recorded on the recording medium, for providing a retrieval key for the unmatched portion.

28. (Original) The computer program product of claim 27 wherein a free memory block of a file computer program product has a block size at least as large as a maximum memory block size.
29. (Original) The computer program product of claim 27 wherein means, recorded on the recording medium, for storing the unmatched portion semi-persistently in the free memory block comprises means, recorded on the recording medium, for storing the unmatched portion without recording the use of the free memory block in the file computer program product.
30. (Original) The computer program product of claim 27 wherein:

means, recorded on the recording medium, for identifying a free memory block of a file computer program product comprises means, recorded on the recording medium, for reading a block identification from a free block list of a file computer program product, and

AUS920030604US1

means, recorded on the recording medium, for storing the unmatched portion semi-persistently in the free memory block comprises means, recorded on the recording medium, for leaving the block identification unchanged in the free block list of the file computer program product.

31. (Cancelled)

32. (Cancelled)

33. (Original) The computer program product of claim 27 wherein means, recorded on the recording medium, for storing the unmatched portion of the chunk comprises means, recorded on the recording medium, for storing the unmatched portion of the chunk as a new memory block having a memory block size equal to the size of the unmatched portion of the chunk.

34. (Original) The computer program product of claim 27 wherein means, recorded on the recording medium, for searching for a segment of the chunk that matches the memory block fails to find a matching segment, the computer program product further comprising means, recorded on the recording medium, for repeatedly carrying out the following steps for all memory blocks in computer memory until a matching segment is found:

retrieving a next memory block from computer memory; and

searching for a segment of the chunk that matches the next memory block.

35. (Original) The computer program product of claim 34 further comprising:

means, recorded on the recording medium, for identifying a free memory block of a file computer program product;

AUS920030604US1

means, recorded on the recording medium, for storing a search section of the chunk semi-persistently in the free memory block; and

means, recorded on the recording medium, for providing a retrieval key for the search section of the chunk.

36. (Original) The computer program product of claim 34 wherein means, recorded on the recording medium, for storing a search section of the chunk comprises means, recorded on the recording medium, for storing the search section of the chunk as a new memory block having a memory block size equal to the size of the search section of the chunk.

37. (Original) The computer program product of claim 34 wherein means, recorded on the recording medium, for providing a retrieval key for a search section of a chunk comprises:

means, recorded on the recording medium, for calculating a weak checksum for the search section of the chunk; and

means, recorded on the recording medium, for calculating a strong checksum for the search section of the chunk.

38. (Original) The computer program product of claim 27 further comprising:

means, recorded on the recording medium, for receiving a retrieval key;

means, recorded on the recording medium, for identifying a memory block in dependence upon the retrieval key;

AUS920030604US1

means, recorded on the recording medium, for retrieving the identified memory block; and

means, recorded on the recording medium, for verifying the contents of the memory block.

39. (Original) The computer program product of claim 38 wherein the retrieval key for the memory block comprises a unique key calculated with an algorithm that generates a unique key from the contents of a memory block, and means, recorded on the recording medium, for verifying the contents of the memory block further comprises:

means, recorded on the recording medium, for calculating a new key for the memory block with the same algorithm; and

means, recorded on the recording medium, for comparing the retrieval key and the new key.